A Parallel Adaptive GA for Linkage Disequilibrium in Genomics.

Laetitia Jourdan, Clarisse Dhaenens, El-Ghazali Talbi

A Parallel adaptive GA for linkage disequilibrium in genomics

Laetitia Vermeulen-Jourdan* Clarisse Dhaenens
El-Ghazali Talbi
LIFL, Bâtiment M3, Cité scientifique, 59655 Villeneuve d'Ascq Cedex - France
{jourdan,dhaenens,talbi}@lifl.fr

Abstract

In this paper, we treat the linkage disequilibrium, used to discover haplotypes, candidate to explain multi-factorial diseases such as diabetes or obesity, as an optimization problem where a given objective function has to be optimized. In order to determine what kind of algorithm will be able to solve this problem, we first study the specificities and the structure of the problem. Results of this study show that exact algorithms are not adapted to this specific problem and lead us to the development of a parallel dedicated adaptive multipopulation genetic algorithm that is able to find several haplotypes of different sizes. After describing the biological problem, we present the dedicated genetic algorithm, its specificities, such as the use of several populations and its advanced mechanisms such as the adaptive choice of operators, random immigrants, and its parallel implementation. We give results on a real dataset.

1 Introduction

The multi-factorial Disease Laboratory of Lille (France) is studying genetic factors that are able to explain diseases such as diabetes or obesity. Some of their experiments deal with linkage disequilibrium mapping, where in order to avoid taking restrictive assumptions, they simultaneously study a very large number of loci on the different chromosomes and look for interesting associations. But the drawback of this approach is the very large number of data generated. Therefore a exploration phase is required to study all this data.
The haplotype analysis by linkage disequilibrium is a challenging way to explore genetic of complex diseases. Data have just started to become available and their studies are quite recent.
Several statistical methods to detect linkage disequilibrium [12, 4, 10] have been used in the past few years. These studies propose statistical models of linkage disequilibrium around a disease susceptibility gene. These methods can only consider associations of one region at a time and make some assumptions on the biological model of the disease.
In the exposed study, biologists want to evaluate haplotypes thanks to statistical computations but they also want to be able to consider associations of several regions. We decide to treat this problem as an optimization one where the objective is to find the best haplotypes in regards with a statistical evaluation.
The next section of the article will firstly present the biological problem and will secondly expose an interesting measure of the quality of haplotypes. Then, the third part will expose the study of the structure of the problem we made and will explain the choice of the method. The fourth part will present the dedicated genetic algorithm, its structure and its specificities, that have been developed for this problem in order to be adapted to the particular objective function we had to use. Moreover, as the evaluation process is time consuming, we will propose a parallel implementation. Then, the fifth section will present results obtained thanks to this algorithm. Finally the conclusion will give indications about exploitation of results.

2 The biological problem

2.1 Biological background

An allele is an alternative form of a gene (one member of a pair) that is located at a specific position on a specific chromosome. Markers are pieces of DNA which allow to characterize an individual, therefore markers can be used to follow the transmission of a piece of chromosome from a generation to another. In particular they are used to look for genes that control some biological characters that we want to study. The sequencing of these markers on a given locus (site on the genome) can show some variations. The existence of these various forms defines what is called genetic polymorphisms. An interested reader should refer to [1] for more information about markers.

*This work was supported by the genopole of Lille

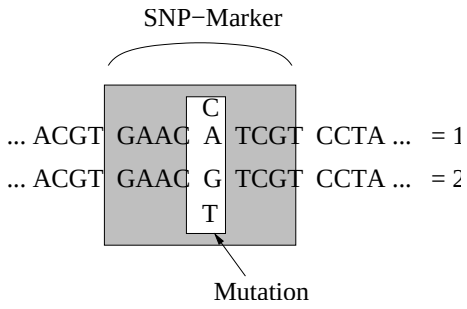

Figure 1. A SNP : an isolated mutation which is grouped into two groups.

A SNP is a biological marker that can have two forms (see figure 1). Classically, one is the original (wild type) and the other is a mutation that occurred n generations ago and was fixed by genetic drift. Most of these mutations are thought to be neutral for selection and for phenotype expression. Among these variations, however, we expect that a minority may have a biological role, especially for the onset of diseases. SNPs loci span all over the genome at irregular distances but with a frequency of 1 every N kb [1].
Much SNP does not have functional implications. Under our genetic model, one allele of a SNP or several alleles of different SNPs, either independently or in combination, increase the risk for the disease (active SNP, SNPa). Then, if we select affected individuals (group A) and unaffected individuals (group U), we will observe a difference in frequencies of alleles of these SNPs between these groups. Curtis and al [3] demonstrate that simultaneous use of several markers is more powerful for identification of chromosome that bears the mutation (SNPa). The problem is however complex as we have no prior knowledge on the number of SNPs involved in the best haplotype nor on the number of active SNPs. SNPs being in the coding areas (SNPc) and in the regulating areas of genes will be particularly interesting to carry out the cartography of the multi-factorial diseases and to study candidate gene associations implied in these diseases.

An haplotype is a set of SNPs (see figure 2). An haplotype allows us to follow simultaneously different genes. We can consider it as a "Meta genome". For example, figure 2 shows an haplotype constituted of 4 SNPs distributed on the genetic card, and it is said that this haplotype has as value 1221 versus 1122 (where a 1 or a 2 represents the 2 possible forms of a SNP). If a SNP is a good indicator to follow a gene, an haplotype can make possible to follow several genes simultaneously [2].
Our goal is to find associations between a status in regards with (Affected / Non Affected for instance) and a multi-locus genotype. This multi-locus genotype is defined by haplotypes.

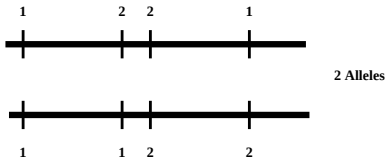

Figure 2. Haplotype 1221/1122.

2.2 Linkage disequilibrium

An association between 2 SNPs at two different loci is called linkage disequilibrium. Association may have different explanations such as population admixture. This correlation reflects the distance between 2 markers and mutation history of the locus in a population. A mutation appears on a chromosome, which has already specific version (alleles) of the existing neighboring SNPs. Then, there will be co-transmission of this new mutation with alleles that were present on this chromosome segment. This correlation can be exploited to identify this locus.
Because of recombinations, associations between the markers and the transfer will decrease from generation to generation and thus *a fortiori* the test of association also, and an imbalance between loci will become less and less detectable. For this reason we will used the haplotypes as more precise markers to follow the transfers.

2.3 Formulation of the problem

The objective of our application is to find haplotypes able to explain the disease under study. These haplotypes may be of different sizes corresponding to the number of SNP that compose the haplotype.
Therefore, in a linkage disequilibrium study, two SNPs of an haplotype must verify the two following conditions:

- their 2 by 2 disequilibrium must be less than a threshold S_1.
- the difference between the smaller frequencies of their 2 variants must be greater than a threshold S_2 .

[1] Unit of length for DNA fragments equal to 1000 nucleotides (Kilobase).

[2] In the case of the study of the multi-factorial diseases, occurrence of a disease depends on several genes.

Generally, there does not exist a single haplotype that verifies all this constraints and allows us to separate affected people with certainty.
So, we will look for several haplotypes of different sizes, that are able to explain the disease under study.

2.4 Evaluation of an haplotype

The objective is to find good associations of SNPs that are able to explain the disease.
In this study, biologists decided to use two procedures, EH-DIALL and CLUMP to evaluate an haplotype. These two procedures, based on statistical computations, are widely used to evaluate haplotypes. The problem we want to solve is not to define a new evaluation scheme of haplotypes, but to generate potential good haplotypes regarding a specific evaluation process.
The problem is to generate potential haplotypes that may be good according to EH-DIALL and CLUMP.
These two procedures are explained below.

2.4.1 EH-DIALL

EH-DIALL[13] (EH is for Estimated Haplotype) is a procedure that determines the most probable distribution of alleles in an haplotype according to values of the SNPs.
Given a sample consisting on a large number of individuals collected at random from the population, EH-DIALL program estimates allele frequencies for each marker. Haplotype frequencies are estimated with allelic association (Hypothesis H_1) and without (Hypothesis H_0).

2.4.2 CLUMP

CLUMP[5] is a program designed to assess the significance of the departure of observed values in a contingency table from the expected values conditional on the marginal totals. CLUMP produces different statistics. The one that corresponds to our problem is referred to as T_1 (χ^2). A good haplotype is an haplotype that is highly correlated with the disease, which corresponds to a high value of T_1.

2.4.3 Evaluation process

Figure 3 represents the whole evaluation process for an haplotype. Starting from a set of candidate SNPs, the process first estimates independently, for affected and unaffected people, the distribution of alleles in the haplotype thanks to EH-DIALL. Then, the procedure CLUMP

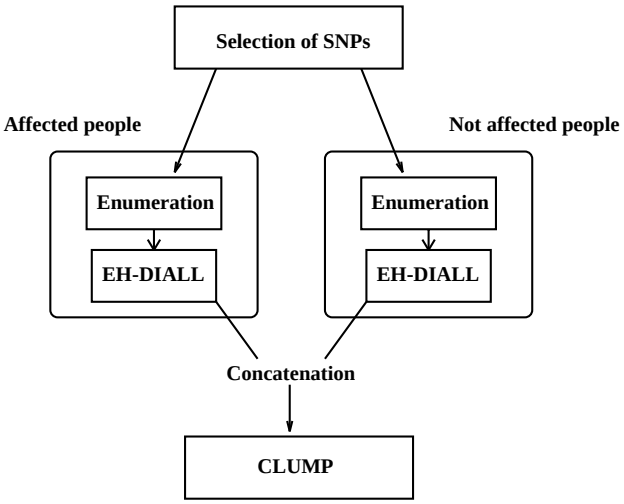

Figure 3. Evaluation of an haplotype.

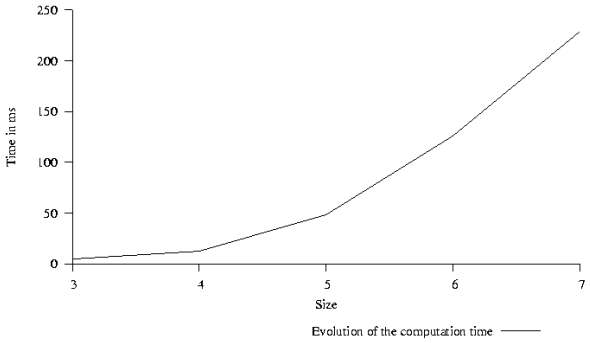

Figure 4. Average time of an evaluation according to the haplotype size.

evaluates the association haplotype-disease.
This process may be very fast for haplotypes composed of 2 or 3 SNPs, but becomes longer when the number of SNPs increases. For example, on a PIV 1.7 Ghz with 256 Mo on a Linux Redhat 8.0 with gcc 3.2, an haplotype of size 3 is evaluated in average time of 6 ms whereas an haplotype of size 7 is evaluated in 201 ms. Figure 4 shows the evolution of the computing time required for the evaluation of associations of different sizes. As we can see, the computing time growths exponentially with the size of haplotypes.

This evaluation process allows to measure in biological terms, the quality of an haplotype. The objective of the search is to find haplotypes that maximize this quality.
Hence, this problem may be considered as an optimization problem, where the search space is composed of all possible associations of SNPs and the criteria to optimize is the result of the evaluation.

3 Choice of the optimization method

The biological problem we have to face consists in finding associations of haplotypes that are able to explain the disease under study.

Optimization methods may be divided into two classes: exact and heuristic methods. When the search space is not too large and the evaluation function not too time consuming, it is possible to enumerate all the solutions and to compare them. This enumeration may use dominance properties in order to avoid enumerating all the solutions but only the most interesting. When, this is not possible, we may use heuristics to deal with this problem. Heuristics may be dedicated to the problem (when it is possible to find particular properties) or may use general schemes (metaheuristics).
When the problem has a very large search space and no evident dominance properties, we may need to use meta-heuristics such as tabu search, simulated annealing or genetic algorithms [14, 11]. An interesting point is that genetic algorithms work on a population of solutions, which may produce, in fine, several solutions to the problem. This is particular interessant in biological problem [9].

In order to choose the method to use, we first study the structure of the problem (landscape). Illustrations are given for a problem with 51 SNPs, because the search space of problems with 150 SNPs, for example, begins to be too large to be completely studied. The currently real data contain at least 249 SNPs for 176 individuals.
Table 1 indicates the number of possible haplotypes of different sizes for problems with 51, 150 SNPs and 249 SNPs. It shows that the search space is very large and impossible to be explored exhaustively. This remark makes the use of an enumeration scheme impossible.

Table 1. Size of the search space

Haplotype size	Number of solutions $C_{\#SNP}^{haplotype\ size}$		
	51 SNPs	150 SNPs	249 SNPs
2	1 275	11 175	30876
3	20 825	551 300	2 542 124
4	249 900	20 260 275	156 340 626
5	2 349 060	591 600 030	$7.6\ 10^9$
6	18 009 460	$14.3\ 10^9$	$3.11\ 10^{11}$

To study the structure of the problem, we have enumerated all possible associations with a small number of SNPs (associations of 2, 3 and 4 SNPs) for a problem with 51 SNPs. For each association, the score (result of the evaluation thanks to EH-DIALL and CLUMP) has been calculated.

The results show different characteristics of the problem:

- First, we can see that some very good haplotypes of size n are not always composed of haplotypes of smaller size with a good score. This characteristic makes the use of constructive method difficult, because this algorithm would combine good haplotypes of size $n-1$ in order to construct haplotypes of size n. With this method it wouldn't be possible to get all the good haplotypes of size n.

- Haplotypes of different sizes are not comparable between them, because the value of their objective function are not in the same range. Indeed, more the haplotype is large more its "value" is large. This aspect eliminates classical enumeration algorithms because they would prefer to construct even larger haplotypes instead of enumerating small sizes haplotypes.

The study of the structure of the problem eliminates some of the optimization approaches. We show that it was important to use a method that is able to deal with a very large search space and that has a good exploration scheme. For these reasons, we decided to develop a genetic algorithm to deal with this problem.

4 A dedicated genetic algorithm

A Genetic Algorithm (GA) works by repeatedly modifying a population of artificial structures through the application of genetic operators [7]. The goal is to find the best

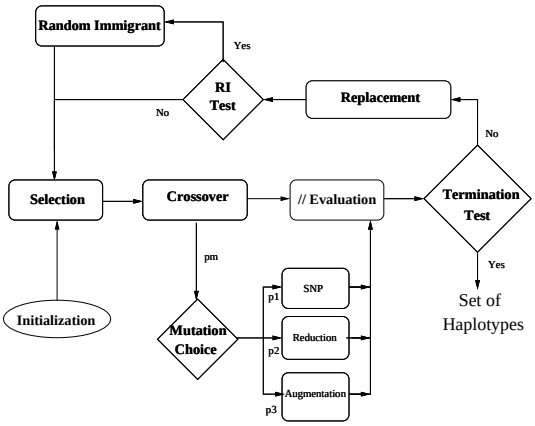

Figure 5. The general scheme of our genetic algorithm.

possible solution or, at least good, solutions for the problem. For this particular biological problem, we have developed a particular GA (see Figure 5). We present here the main characteristics and adaptations that we made to deal with the specific problem, and in particular with the specific evaluation function.

4.1 Encoding

An haplotype is a structure composed of:

- an integer indicating the size t of the haplotype.
- a table with t SNPs ordered in the ascending order without repetition.
- a real to store the value of the individual.

4.2 Population and individuals

One of the major disadvantages of this problem is that haplotypes of different sizes are not directly comparable between them. To overcome this problem, our global population will be divided into several subpopulations, where each subpopulation corresponds to a given size of haplotype. The number of individuals in each subpopulation are not equal and increases with the size of the haplotypes in order to follow the growth of the size of the search space related to each size. Some cooperations will exist between subpopulations.

4.3 Operators

Operators allow GAs to explore the search space. However, operators typically have destructive as well as constructive effects. They must be adapted to the problem.

4.3.1 Mutation

Mutation is an operator which allows diversity. We make three kinds of mutation:

- Mutation of a SNP: we randomly choose a SNP of the individual and replace it by another randomly chosen SNP. This process is similar to a local search which allows to explore the neighborhood of the solution. We use this mutation several times in parallel and keep the best individual found by this mutation.
- Reduction Mutation: we randomly choose a SNP of the individual and remove it. The individual has now a lower size. It allows to move individuals from a subpopulation to another.
- Augmentation Mutation: we add a randomly chosen SNP.

Probabilities of mutation are hard to set when we have several mutation operators and authors often set them experimentally. To overcome this problem, we implement an adaptive strategy for calculating the rate of each mutation operator. In [8], the authors proposed to compute the new rate of mutation by calculating the progress of the j^{th} application of mutation M_i, for an individual ind mutated into an individual mut as follows:

$$progress_j(M_i) = Max(f(ind), f(mut)) - f(ind)$$

But some mutation operators increase or decrease the number of SNPs and we saw that the fitness function used is correlated to the number of SNPs. In order to adapt the notion of progress to our problem, we normalize the progress with the best individual (B) and the worst of the subpopulation (W) corresponding to the individual mut (the best individual of the same size). We define the normalized value of the fitness of an individual ind as:

$$norm(ind) = \frac{f(ind) - W(ind.size)}{B(ind.size) - W(ind.size)}$$

So the progress is now:

$$progress_j(M_i) = Max(norm(ind), norm(mut)) - norm(ind)$$

Then for all the mutations operators M_i, assume $Nb\_mut(M_i)$ applications of the mutation are done at a given generation ($j = 1, .., Nb\_mut(M_i)$). Then we can compute the profit of a mutation operator M_i :

$$Profit(M_i) = \frac{\sum_j progress_j(M_i)/Nb\_mut(M_i)}{\sum_k \left(\sum_j progress_j(M_k)/Nb\_mut(M_k)\right)}$$

We set a minimum rate δ and a global mutation rate $p_{mutation}$ for N mutation operators to apply. The new mutation ratio for each M_i is calculated using the following

formula [8]:

$$p(M_i) = Profit(M_i) \times (p_{mutation} - N \times \delta) + \delta$$

The sum of all the mutation rates is equal to the global rate of mutation $p_{mutation}$. The initial rate of each mutation operator is set to $p_{mutation}/N$.

4.3.2 Crossover

We use an uniform crossover: take the two strings of SNPs of the parents and create two children by randomly shuffling the variables corresponding to the SNP at each site, then eventually send a child to the mutation and update the scores and the numbers of SNPs of the children.

We use two kinds of crossovers:

- Intra-population: only crossovers between individuals of a same subpopulation are allowed
- Inter-population: crossovers between individuals of different subpopulations are allowed, creating one child of each parents size.

The probabilities of each kind of crossover are set adaptively. We adapt the strategy used for our mutations (4.3.1) to the case of the quadratic crossover. We define the average improvement of a child e with regards to its parents p_1 and p_2 for intra-population crossover as:

$$Improve_{Intra}(e, p_1, p_2) = \left(\begin{array}{l} \frac{Max(f(p_1), f(e)) - f(p_1)}{B(e.size)} \\ + \\ \frac{Max(f(p_2), f(e)) - f(p_2)}{B(e.size)} \end{array} \right) /2$$

In this case, the three individuals e, p_1 and p_2 are of the same size. They can be compared.

We define the improvement for the inter-population crossover by only comparing the improvement between a child e and its parent of the same size:

$$Improve_{Inter}(e, p_1, p_2) =$$

$$\left\{ \begin{array}{ll} \frac{Max(f(p_1), f(c)) - f(p_1)}{B(e.size)} & \text{If } size.p_1 = size.e \\ \frac{Max(f(p_2), f(c)) - f(p_2)}{B(e.size)} & \text{If } size.p_2 = size.e \end{array} \right.$$

So the global function for the improvement is:

$$Improve(e, p_1, p_2) = \left\{ \begin{array}{c} Improve_{Intra}(e, p_1, p_2) \\ \text{OR } Improve_{Inter}(e, p_1, p_2) \end{array} \right.$$

The progress of the j^{th} application of each crossover C_i, which mates two individuals p_1 and p_2 to obtain two children e_1 and e_2 is:

$$progress_j(C_i) = \left\{ \begin{array}{c} Improve_j(e_1, p_1, p_2) + \\ Improve_j(e2, p_1, p_2) \end{array} \right.$$

Then for all the crossover operators C_i, assume $Nb\_cross(C_i)$ applications of the crossover are made during a given generation. Then the profit of a crossover operator C_i is:

$$Profit(C_i) = \frac{\sum_j progress_j(C_i)/Nb\_cross(C_i)}{\sum_k \left(\sum_j progress_j(C_k)/Nb\_cross(C_k) \right)}$$

We set a minimum rate δ and a global mutation rate $p_{crossover}$ for N crossover operators to apply. The new crossover ratio for each C_i is calculated using the following formula [8]:

$$p(C_i) = Profit(C_i) \times (p_{crossover} - N \times \delta) + \delta$$

The sum of all the crossover rate is equal to the global rate of crossover $p_{crossover}$. The initial rate of each crossover operator is set to $p_{crossover}/N$.

4.4 Random Immigrant

As the search space is very large, it is important to have a wide exploration. Random Immigrant is another process that helps to maintain diversity in the population by introducing new individuals [2]. It should also help to avoid premature convergence. We use random immigrant as follows (see figure 5): when the best individual is the same during N generations, all the individuals of the population, whose scores are under the mean, are replaced by new individuals randomly generated.

4.5 Parallel implementation

We saw during our experiment that the evaluation function can be time consuming (see figure 4). In order to run the algorithm in a reasonable time, we have made a synchronous parallel implementation of the evaluation phase.
The implementation is based on a master / slaves model (see figure 6). The slaves are initiated at the beginning and access only once to the data. During the evaluation phase, the master gives each slave an individual to evaluate. Then the slave computes the fitness of this individual and send it back to the master.
The programming environment used is C/PVM (Parallel Virtual Machine) [6].

4.6 Replacement and termination test

The replacement of an individual is very simple. A new individual is inserted in the current population if it is better than the worst individual of the population and if it is not already in the population.
In order to let the algorithm progress, we decide to stop the

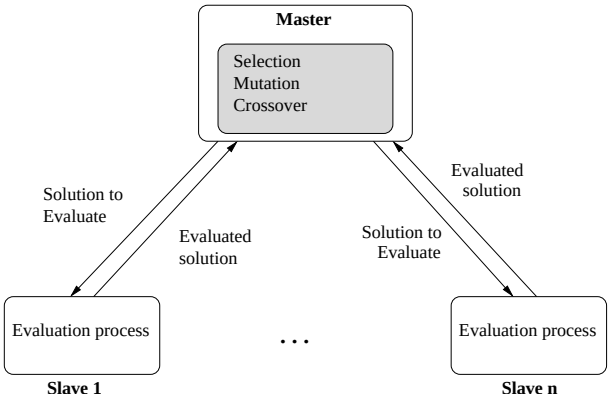

Figure 6. Synchronous master / slaves model for parallel genetic algorithm.

algorithm when the best individual has not evolved during a fixed number of generations. That is why our algorithm has not always the same number of generations.

5 Experiments

We execute our method on data provided by the Biological Institute of Lille which are related to diabetes/obesity. The data set contains 176 individuals : 53 affected individuals, 53 healthy individuals, 70 unknown. Data are composed of different tables. The study reported here is composed of 106 individuals and 51 SNPs.
Other experiments, but not so complete have been done with larger files (249 SNPs).

5.1 Data

A first table gives information about people that belongs to the two groups, affected and healthy individuals. This table indicates the values of SNPs for all the people.

Two other tables give information on SNPs themselves. They are necessary, firstly for EH-DIALL and CLUMP algorithms and, secondly to verify if an haplotype respects the two conditions on the frequency and the disequilibrium between SNPs (see 2.3). A table indicates for each SNP the frequency of each alternative (1 and 2). The last table gives the disequilibrium between every couples of SNPs.

5.2 Results

We tested the genetic algorithm proposed in different manners in order to find the best configuration. Thus, we tested the following schemes :

- Without and with the random immigrant.
- Without and with the reduction and the augmentation mutation
- Without and with the inter-population crossover

It appeared that mechanisms that link subpopulations are efficient and allow to find better solutions than without them. The random immigrant is able to introduce diversity, when the search seems to be blocked.

5.2.1 Parameters

A genetic algorithm has several parameters and the user can set the maximal size of an haplotype. Biologists choose 6 for this size as a first experiment. For our experiments, we set:

- $P_{mutation}$=0.9
- $\delta = 0.1$
- Population Size = 150
- Number of generations where the best is the same= 100
- Max. Size of an haplotype = 6
- Random Immigrant stagnation : 20

5.2.2 Results

Table 2 presents results obtained over 10 runs with the best combination of our mechanisms. We indicate the mechanisms used in the column “Scheme”, the best haplotype found (over 10 runs) for each subpopulation, its size and its fitness. Then we report the mean fitness obtained over the ten runs and the mean difference, called deviation (Dev.) with the best expected haplotype. Moreover, we present the number of evaluations required to obtain the solution. We put the minimum found over the 10 runs and the mean.

Let’s note that the introduction of advanced mechanisms requires additional computations. But in our case, the evaluation is costly, so an interesting indicator is the number of evaluations needed. Concerning the number of evaluations required to obtain solutions, let’s refer to table 1 (Size of the space search). The scheme implemented is able to find the best solutions while exploring a very small part of the search space.

This algorithm has proved to be able to provide good solutions for the problem of selecting interesting associations of SNPs. In the case of 51 SNPs, we could compare the exact solutions obtained with the best solutions calculated

Table 2. Results obtained by the GA for 51 SNPs.

Scheme	Best Haplotype	Fitness	Mean	Dev	Min # of Eval.	Mean
Adaptive Mutation	8 12 15	58.814	58.814	0	317	587.4
+ Adaptive crossover	8 18 26 50	84.856	84.856	0	1111	3238.2
+ Random Immigrant	8 12 16 33 43	123.108	123.108	0	2994	5615.2
	8 12 15 21 32 43	161.252	161.252	0	11573	15464.6

during the study of landscape. The genetic algorithm finds, most of time, the different best solutions. On larger problems, for example a real data set of 249 SNPs, it has shown a good robustness (solutions provide are similar from one execution to another). This algorithm is used by the biologists in an extensive manner.

6 Conclusion

This article presents an optimization approach to deal with a linkage disequilibrium study. We explain the choice of the method and the elimination of classical algorithms. We expose the specific multipopulation genetic algorithm we developed. Results obtained with this algorithm allow to extract interesting associations of SNPs regarding to the evaluation function given by the biologists.

The time complexity of the evaluation function make necessary to adopt a parallel strategy in order to have results in reasonable time.

Thanks to this method, biologists are able to test different data sets and to formulate hypothesis on genetic factors involved in diseases under study.

Moreover, different objective functions are going to be used in order to compare them and to validate their biological interest.

In this work we applied an optimization strategy, while studying the structure of the problem before choosing the method to use, to a specific problem that was first not exposed by biologists as an optimization problem. Such a strategy could be applied for other problems for which the search space may be described and an evaluation function may be defined, in order to develop well adapted methods.

References

[1] D. Boichard, P. L. Roy, H. Levéziel, and J.-M. Elsen. Utilisation des marqueurs moléculaires en génétique animale. *INRA Production Animale*, pages 67–80, 1998.

[2] C. B. Congdon. *A comparison of genetic algorithm and other machine learning systems on a complex classification task from common disease research*. PhD thesis, University of Michigan, 1995.

[3] D. Curtis, B. North, and P. Sham. Use of an artificial neural network to detect association between a disease and multiple marker genotypes. *Ann Hum Genet*, 65(1):95–107, 2001.

[4] B. Devlin, N. Risch, and K. Roeder. Disequilibrium mapping: Composite likelihood for pairwise disequilibrium. *Genomics*, 36:1–16, 1996.

[5] P. S. et D. Curtis. Monte carlo tests for associations between disease and alleles at hightly polymorphic loci. *Annal Human Genetic*, pages 97–105, 1995.

[6] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Mancbek, and V. Sunderam. *PVM: Parallel Virtual Machine - A User's Guide and Tutorial for Networked Parallel Computing,*. MIT PRess, 1994.

[7] D. E. Goldberg. *Genetic Algorithms - in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, 1989.

[8] T. P. Hong, H. Wang, and W. Chen. Simultaneosly applying multiple mutation operators in genetic algorithms. *Journal of Heuristics*, 6:439 – 455, 2000.

[9] L. Jourdan, C. Dhaenens, and E.-G. Talbi. *Evolutionary Computation in Bioinformatic.*, chapter Discovery of Genetic and Environmental Interactions in Disease Data using Evolutionary Computation, pages 297–316. Morgan Kaufmann, 2002.

[10] L. Lazzeroni. Linkage disequilibrium and gene mapping: an empirical least-squares approach. *Am. Journal Human Genetic*, 62:159–170, 1998.

[11] M. Pei, E. Goodman, and W. Punch. Feature extraction using genetic algorithms. Technical report, Michigan State University : GARAGe, June 1997.

[12] J. Terwilliger. A powerful likelihood method for the analysis of linkage disequilibrium between trait loci and one or more polymorphic marker loci. *Am J Hum Genet*, 1995.

[13] J. Terwilliger and J. Ott. *Handbook of human genetic linkage*. Johns Hopkins University Press, Baltimore, June 1994. ISBN: 0801848032.

[14] J. Yang and V. Honoavar. *Feature Extraction, Construction and Selection : A data Mining Perspective*, chapter 1: Feature Subset Selection Using a Genetic Algorithm, pages 117–136. H. Liu and H. Motoda Eds, massachussetts : kluwer academic publishers edition, 1998.